PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: [✓ The ACM Digital Library] The Guide

fixed length string and data string

SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **fixed** **length** **string** and **data** **string** Found **62,179** of **147,060**

Sort results by	relevance
Display results	expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 String storage and searching for data base applications: Implementation on the INDY backend kernel
George P. Copeland
August 1978 **Proceedings of the fourth workshop on Computer architecture for non-numeric processing**

Full text available: pdf(854.23 KB) Additional Information: full citation, abstract, references, citings, index terms

User and hardware cost trends dictate that data base systems should provide more complete functionality, simplicity of use, and reliability by increasing the amount of hardware present in the system. These goals are accomplished with a simple hardware arrangement within a one-dimensional cellular storage system called INDY. The INDY backend kernel is intended as a powerful tool for implementing all data models. The INDY cellular storage array is intended to provide functionality that is dif ...

2 String storage and searching for data base applications: implementation on the INDY backend kernel
George P. Copeland
August 1978 , Volume 10 , 13 , 7 Issue 1 , 2 , 2

Full text available: pdf(986.51 KB) Additional Information: full citation, abstract, references

User and hardware cost trends dictate that data base systems should provide more complete functionality, simplicity of use, and reliability by increasing the amount of hardware present in the system. These goals are accomplished with a simple hardware arrangement within a one-dimensional cellular storage system called INDY. The INDY backend kernel is intended as a powerful tool for implementing all data models. The INDY cellular storage array is intended to provide functionality that is difficul ...

3 Definable relations and first-order query languages over strings
Michael Benedikt, Leonid Libkin, Thomas Schwentick, Luc Segoufin
September 2003 **Journal of the ACM (JACM)**, Volume 50 Issue 5

Full text available: pdf(587.44 KB) Additional Information: full citation, abstract, references, index terms

We study analogs of classical relational calculus in the context of strings. We start by studying string logics. Taking a classical model-theoretic approach, we fix a set of string operations and look at the resulting collection of definable relations. These form an algebra--a class of *n*-ary relations for every *n*, closed under projection and Boolean operations. We show that by choosing the string vocabulary carefully, we get string logics that have desirable properties: computable ...

Keyw rds: Strings, expressive power, first-order definability, quantifier elimination, query languages

4 Design and characteristics of a variable-length record sort using new fixed-length record sorting techniques
Martin A. Goetz
May 1963 **C mmunicati ns f the ACM**, Volume 6 Issue 5

Full text available: pdf(358.54 KB) Additional Information: full citation, abstract, references, citings

This paper describes the application of several new techniques for sorting fixed-length records to the problem of variable-length record sorting. The techniques have been implemented on a Sylvania 9400 computer system with 32,000 fixed-length words of memory. Specifically, the techniques sequence variable-length records of unrestricted size, produce long initial strings of data, merge strings of data at the power of T - 1, where T is the number of work tape ...

5 Burst tries: a fast, efficient data structure for string keys
April 2002 **ACM Transactions on Information Systems (TOIS)**, Volume 20 Issue 2

Full text available: pdf(324.84 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many applications depend on efficient management of large sets of distinct strings in memory. For example, during index construction for text databases a record is held for each distinct word in the text, containing the word itself and information such as counters. We propose a new data structure, the burst trie, that has significant advantages over existing options for such applications: it uses about the same memory as a binary search tree; it is as fast as a trie; and, while not as fast as a ...

Keywords: Binary trees, splay trees, string data structures, text databases, tries, vocabulary accumulation

6 The string B-tree: a new data structure for string search in external memory and its applications
Paolo Ferragina, Roberto Grossi
March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(363.37 KB) Additional Information: full citation, abstract, references, citings, index terms

We introduce a new text-indexing data structure, the String B-Tree, that can be seen as a link between some traditional external-memory and string-matching data structures. In a short phrase, it is a combination of B-trees and Patricia tries for internal-node indices that is made more effective by adding extra pointers to speed up search and update operations. Consequently, the String B-Tree overcomes the theoretical limitations of inverted files, B-trees, prefix B-trees, s ...

Keywords: B-tree, Patricia trie, external-memory data structure, prefix and range search, string searching and sorting, suffix array, suffix tree, text index

7 Performance of data structures for small sets of strings
Steffen Heinz, Justin Zobel
January 2002 **Australian Computer Science C mmunicati ns , Pr ceedings f the twenty-fifth Australasian c nference n Computer science - V lume 4**, Volume 24 Issue 1

Full text available: pdf(929.27 KB) Additional Information: full citation, abstract, references, citings, index terms

Fundamental structures such as trees and hash tables are used for managing data in a huge variety of circumstances. Making the right choice of structure is essential to efficiency. In previous work we have explored the performance of a range of data structures---different forms of trees, tries, and hash tables---for the task of managing sets of millions of strings, and have developed new variants of each that are more efficient for this task than previous alternatives. In this paper we test the ...

Keyw rds: binary search tree, burst trie, data structures, inverted index, splay tree, trie

8 A guided tour to approximate string matching
Gonzalo Navarro
March 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 1

Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We survey the current techniques to cope with the problem of string matching that allows errors. This is becoming a more and more relevant issue for many fast growing areas such as information retrieval and computational biology. We focus on online searching and mostly on edit distance, explaining the problem and its relevance, its statistical behavior, its history and current developments, and the central ideas of the algorithms and their complexities. We present a number of experiments to ...

Keywords: Levenshtein distance, edit distance, online string matching, text searching allowing errors

9 Algorithmic selection of the best method for compressing map data strings
E. L. Amidon, G. S. Akin
December 1971 **Communications of the ACM**, Volume 14 Issue 12

Full text available: pdf(445.93 KB) Additional Information: full citation, abstract, references, citings

The best of a dozen different methods for compressing map data is illustrated. The choices are generated by encoding data strings—sequence of like codes—by three methods and in four directions. Relationships are developed between compression alternatives to avoid comparing all of them. The technique has been used to compress data from forest resource maps, but is widely applicable to map and photographic data reduction.

Keywords: data compression, data reduction, information retrieval, input/output, map storage, run coding

10 Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract)
Roberto Grossi, Jeffrey Scott Vitter
May 1999 **Proceedings of the thirty-second annual ACM symposium on Theory of computing**

Full text available: pdf(1.11 MB) Additional Information: full citation, references, citings, index terms

11 Linear Algorithm for Data Compression via String Matching
Michael Rodeh, Vaughan R. Pratt, Shimon Even
January 1981 **Journal of the ACM (JACM)**, Volume 28 Issue 1

Full text available: pdf(507.76 KB) Additional Information: full citation, references, citings, index terms

12 On effective multi-dimensional indexing for strings
H. V. Jagadish, Nick Koudas, Divesh Srivastava
May 2000 **ACM SIGMOD Record , Proceedings f the 2000 ACM SIGMOD internati nal**

conference n Management f data, Volume 29 Issue 2

Full text availabl : pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

As databases have expanded in scope from storing purely business data to include XML documents, product catalogs, e-mail messages, and directory data, it has become increasingly important to search databases based on wild-card string matching: prefix matching, for example, is more common (and useful) than exact matching, for such data. In many cases, matches need to be on multiple attributes/dimensions, with correlations between the dimensions. Traditional multi-dimensional index structures, ...

13 Data compression
Debra A. Lelewer, Daniel S. Hirschberg
September 1987 **ACM Computing Surveys (CSUR)**, Volume 19 Issue 3

Full text available: pdf(3.61 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper surveys a variety of data compression methods spanning almost 40 years of research, from the work of Shannon, Fano, and Huffman in the late 1940s to a technique developed in 1986. The aim of data compression is to reduce redundancy in stored or communicated data, thus increasing effective data density. Data compression has important application in the areas of file storage and distributed systems. Concepts from information theory as they relate to the goals and evaluation of data ...

14 Data compression with finite windows
E. R. Fiala, D. H. Greene
April 1989 **Communications of the ACM**, Volume 32 Issue 4

Full text available: pdf(1.89 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Several methods are presented for adaptive, invertible data compression in the style of Lempel's and Ziv's first textual substitution proposal. For the first two methods, the article describes modifications of McCreight's suffix tree data structure that support cyclic maintenance of a window on the most recent source characters. A percolating update is used to keep node positions within the window, and the updating process is shown to have constant amortized cost. Other methods explore the ...

15 Sequences and strings: On effective classification of strings with wavelets
Charu C. Aggarwal
July 2002 **Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining**

Full text available: pdf(889.02 KB) Additional Information: full citation, abstract, references, citings, index terms

In recent years, the technological advances in mapping genes have made it increasingly easy to store and use a wide variety of biological data. Such data are usually in the form of very long strings for which it is difficult to determine the most relevant features for a classification task. For example, a typical DNA string may be millions of characters long, and there may be thousands of such strings in a database. In many cases, the classification behavior of the data may be hidden in the comp ...

16 Fast string searching in secondary storage: theoretical developments and experimental results
Paolo Ferragina, Roberto Grossi
January 1996 **Proceedings f the seventh annual ACM-SIAM symp sium n Discrete alg rithms**

Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

17 Finite state machines for strings over infinite alphabets
Frank Neven, Thomas Schwentick, Victor Vianu
July 2004 **ACM Transacti ns n C mputati nal L gic (TOCL)**, Volume 5 Issue 3
Full text available: pdf(346.72 KB) Additional Information: full citation, abstract, ref rences, index terms

Motivated by formal models recently proposed in the context of XML, we study automata and logics on strings over infinite alphabets. These are conservative extensions of classical automata and logics defining the regular languages on finite alphabets. Specifically, we consider register and pebble automata, and extensions of first-order logic and monadic second-order logic. For each type of automaton we consider one-way and two-way variants, as well as deterministic, nondeterministic, and alterna ...

Keywords: Automata, XML, expressiveness, first-order logic, infinite alphabets, monadic second-order logic, pebbles, registers

18 String processing techniques
Stuart E. Madnick
July 1967 **Communications of the ACM**, Volume 10 Issue 7
Full text available: pdf(714.09 KB) Additional Information: full citation, abstract, references, citings, index terms

The internal organization of string processing systems is discussed. Six techniques for data structures are presented and evaluated on the basis of: (1) creation of strings; (2) examination of strings; and (3) alteration of strings. Speed of operation, storage requirements, effect on paging, and programmer convenience are also considered. One of the techniques, single-word linked blocks, is used in an example demonstrating an implementation of a SNOBOL string processing language on an IBM S ...

19 Dictionary-based order-preserving string compression
Gennady Antoshenkov
February 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 1
Full text available: pdf(203.08 KB) Additional Information: full citation, abstract, index terms

As no database exists without indexes, no index implementation exists without order-preserving key compression, in particular, without prefix and tail compression. However, despite the great potentials of making indexes smaller and faster, application of general compression methods to ordered data sets has advanced very little. This paper demonstrates that the fast dictionary-based methods can be applied to order-preserving compression almost with the same freedom as in the general case. The pro ...

Keywords: Indexing, Order-preserving key compression

20 A new string search hardware architecture for VLSI
K. Takahashi, H. Yamada, H. Nagai, K. Matsumi
June 1986 **ACM SIGARCH Computer Architecture News , Proceedings of the 13th annual international symposium on Computer architecture**, Volume 14 Issue 2
Full text available: pdf(683.39 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a new architecture for practical string search hardware design. This architecture is based on the finite state automaton design concept using a character control charge transfer model. The resultant hardware is a set of programmable sequential logic (PSL) circuits, each of which consists of a sequential logic and memory parts. The logic part is an array of logical gates, each of which is controlled by the read-out signal from the memory part, to connect the flip-flops. T ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next